

IN THE CLAIMS

Please amend the claims as follows:

Claims 1-25 (Canceled).

Claim 26 (New): A task execution method of allocating a plurality of tasks to a plurality of processors and then causing the processors to execute the tasks, the tasks composing a program and each of the tasks being described using any one of different instruction sets of the processors, the method comprising:

a first operation of allocating all of the tasks to the processors such that each of the tasks is allocated to a processor having an instruction set that is equal to an instruction set used for describing said each of the tasks;

a second operation of sequentially executing processes to determine whether the program is improved in execution efficiency when an allocation destination of each of the tasks is changed from the processor to which said each of the tasks is allocated in the first operation to another processor;

a third operation of changing an allocation destination of a task that is determined to be one improving in execution efficiency to said another processor; and

a fourth operation of causing the processors to execute all of the tasks after the third operation.

Claim 27 (New): The method according to claim 26, wherein the tasks mutually have an inter-task dependency.

Claim 28 (New): The method according to claim 26, wherein the second operation includes:

estimating a data amount in inter-processor communications in an entire program when an allocation destination of each of the tasks is changed to another processor and a data amount in inter-processor communications in the entire program when the allocation destination is not changed; and

determining whether the program is improved in execution efficiency when the allocation destination of said each of the tasks is changed to said another processor according to whether the data amount in inter-processor communications in the entire program decreases.

Claim 29 (New): The method according to claim 26, wherein said second operation includes sequentially performing:

a process of determining whether one of tasks immediately before and after each of the tasks is a to-be-optimized task that is allocated to a processor different from the processor to which said each of the tasks is allocated in the first operation; and

a process of determining whether the program is improved in execution efficiency when allocation destinations of tasks that are determined to be to-be-optimized tasks are changed from the processor to which said each of the tasks is allocated in the first operation to another processor.

Claim 30 (New): A multiprocessor system including a plurality of processors including different instruction sets, comprising:

a provisional allocation device that provisionally allocates all of a plurality of tasks to the processors such that each of the tasks is allocated to a processor having an instruction set

that is equal to an instruction set used for describing said each of the tasks, the tasks composing a program and each of the tasks being described using any one of the different instruction sets;

a determination device that sequentially executes processes to determine whether the program is improved in execution efficiency when an allocation destination of each of the tasks is changed from the processor to which said each of the tasks is allocated to another processor; and

an allocation destination changing device that changes an allocation destination of a task that is determined to be one improving in execution efficiency to said another processor, wherein when the allocation destination changing device changes allocation destinations of tasks that are determined to be ones improving in execution efficiency, the processors execute the tasks allocated thereto.

Claim 31 (New): The system according to claim 30, wherein the tasks mutually have an inter-task dependency.

Claim 32 (New): The system according to claim 30, wherein the determination device:

estimates a data amount in inter-processor communications in an entire program when an allocation destination of each of the tasks is changed to another processor and a data amount in inter-processor communications in the entire program when the allocation destination is not changed; and

determines whether said each of the tasks is improved in execution efficiency when the allocation destination of said each of the tasks is changed to said another processor

according to whether the data amount in inter-processor communications in the entire program decreases.

Claim 33 (New): The system according to claim 30, wherein the determination device sequentially performs:

a process of determining whether one of tasks immediately before and after each of the tasks is a to-be-optimized task that is provisionally allocated to a processor different from the processor to which said each of the tasks is provisionally allocated by the provisional allocation device; and

a process of determining whether the program is improved in execution efficiency when allocation destinations of tasks that are determined to be to-be-optimized tasks are changed from the processor to which said each of the tasks is provisionally allocated to another processor.

Claim 34 (New): A computer readable storage medium storing computer executable instructions of a computer program for causing a computer to execute a process of allocating a plurality of tasks to a plurality of processors and then executing the tasks, the tasks composing a program and each of the tasks being described using any one of different instruction sets of the processors, which when executed by the computer results in performance of operations comprising:

a first operation of allocating all of the tasks to the processors such that each of the tasks is allocated to a processor having an instruction set that is equal to an instruction set used for describing said each of the tasks;

a second operation of sequentially executing processes to determine whether the program is improved in execution efficiency when an allocation destination of each of the

tasks is changed from the processor to which said each of the tasks is allocated in the first operation to another processor;

a third operation of changing an allocation destination of a task that is determined to be one improving in execution efficiency to said another processor; and

a fourth operation of causing the processors to execute all of the tasks after the third operation.

Claim 35 (New): The medium according to claim 34, wherein the tasks mutually have an inter-task dependency.

Claim 36 (New): The medium according to claim 34, wherein the second operation includes:

estimating a data amount in inter-processor communications in an entire program when an allocation destination of each of the tasks is changed to another processor and a data amount in inter-processor communications in the entire program when the allocation destination is not changed; and

determining whether the program is improved in execution efficiency when the allocation destination of said each of the tasks is changed to said another processor according to whether the data amount in inter-processor communications in the entire program decreases.

Claim 37 (New): The medium according to claim 34, wherein the second operation includes sequentially performing:

a process of determining whether one of tasks immediately before and after each of the tasks is a to-be-optimized task that is allocated to a processor different from the processor to which said each of the tasks is allocated in the first operation; and

a process of determining whether the program is improved in execution efficiency when allocation destinations of tasks that are determined to be to-be-optimized tasks are changed from the processor to which said each of the tasks is allocated in the first operation to another processor.